Test-Time 3D Occupancy Prediction

Supplementary Material

A. Implementation Details

A.1. Depth Estimation and Triangulation-Based Calibration for TT-OccCamera

When the outputs of 3DVFMs are already in a metric scale (such as MapAnything [6], they can be directly applied. However, for models that do not provide metric outputs (such as VGGT [8]), additional calibration is required to align them with the ground-truth occupancy. Using VGGT as an example, we describe below the depth estimation process and a triangulation-based calibration method. VGGT is a feed-forward neural network capable of predicting depth maps and tracking 2D keypoints across frames. We input six surrounding camera views into VGGT to generate per-view depth predictions. Following the original VGGT setup, the input images are resized to a resolution of 294×518. Although VGGT produces consistent and high-quality depth estimates across views, the predictions are in an unscaled unit space and do not correspond directly to real-world metric distances. To address this limitation, we leverage VGGT's built-in 2D point tracking functionality across multiple views at the same time step. Specifically, we select three adjacent cameras including front, front-left, and front-right, and use VGGT to track sparse 2D keypoints across them. By filtering out low-quality matches using the predicted visibility and confidence scores, we obtain reliable point correspondences between camera pairs, as illustrated in Fig. 1. We then triangulate these matched 2D points using the ground-truth camera intrinsics and extrinsics provided by the dataset, resulting in a sparse but metrically accurate 3D point cloud. Finally, we compare the magnitudes of the triangulated 3D points with those reconstructed from the predicted depth maps at the corresponding image locations, and compute a global scaling factor to align the depth predictions with real-world scale. An example of the final scaled depth prediction is shown in Fig. 2.

A.2. Open-Vocabulary Semantic Segmentation

We now describe the process of open-vocabulary semantic segmentation using OpenSeeD [9] as an example. We adopt OpenSeeD primarily to ensure a fair comparison with SelfOcc [4]. Nevertheless, our system is loosely coupled with VFMs and fully compatible with more advanced ones, such as REX-Omni [5]. As shown in Fig. 3, OpenSeeD's predictions often exhibit noisy and unclear boundaries. Since our focus is not prompt engineering and to ensure a fair comparison with SelfOcc [4], we adopt the same query set including: *"barrier"*, *"bicycle"*, *"bus"*, *"car"*, *"construction_vehicle"*, *"crane"*, *"motorcycle"*, *"person"*, *"traffic_cone"*, *"trailer"*, *"trailer_truck"*, *"truck"*, *"road"*, *"sidewalk"*, *"terrain"*, *"grass"*, *"building"*, *"wall"*, *"tree"*, *"sky"*.

A.3. Tracking with RAFT for TT-OccCamera

For TT-OccCamera, we estimate the optical flow F_{opt} between two consecutive frames from the same camera using RAFT [7]. We then compute the ego-motion-induced flow F_{ego} based on the ground-truth camera intrinsics and extrinsics of the adjacent frames, along with the predicted depth from 3DVFMs. By subtracting the ego flow from the observed optical flow, we obtain the dynamic flow $F_{\text{dyn}} = F_{\text{opt}} - F_{\text{ego}}$, which theoretically captures the motion of dynamic objects in the environment. Although this 2D dynamic flow could, in principle, guide the 3D motion of dynamic Gaussians, back-projecting it into 3D space tends to amplify errors from RAFT and 3DVFMs, resulting in unstable Gaussian motion. To mitigate this, we adopt a compromise strategy by thresholding the dynamic flow magnitude to obtain a dynamic mask that identifies likely moving regions. In the ideal case, a simple thresholding on the magnitude of F_{dyn} would yield a reliable binary mask for dynamic regions. However, since both F_{opt} and F_{ego} are derived from 2D estimations and are subject to noise and inaccuracies, the resulting F_{dyn} is often highly unreliable and noisy. To further refine the dynamic flow, we leverage the the cues from semantic segmentation models, which provides relatively cleaner object boundaries, to refine the dynamic flow magnitude map. As illustrated in Fig. 4, the raw dynamic flow is noisy, and thresholding it directly often produces fragmented masks that do not correspond to coherent objects. After incorporating instance masks from segmentation models, high-magnitude errors on the background are suppressed, and the resulting dynamic masks become more object-aligned, either an entire object is identified as dynamic or it is not, effectively eliminating partial or spurious activations. The corresponding 3D Gaussians projected onto these regions are treated as dynamic and excluded from static accumulation in the next frame. While this approach does not allow accumulation of dynamic objects as in the LiDAR-based variant, it effectively reduces artifacts caused by noisy motion cues and temporal inconsistencies.

A.4. Tracking with LiDAR for TT-OccLiDAR

Tracking in TT-OccLiDAR is generally more reliable than in TT-OccCamera, as LiDAR point clouds provide more accurate and consistent geometric information. We follow a straightforward strategy: cluster first, then align via ICP. First, we project LiDAR points onto the instance masks predicted by the segmentation model, thereby associating

Figure 1. Visualization of VGGT-predicted 2D tracking across front-left, front, and front-right cameras. Sparse query points are tracked and subsequently triangulated to obtain a metric 3D point cloud, which is used to align the predicted depth maps to real-world scale.

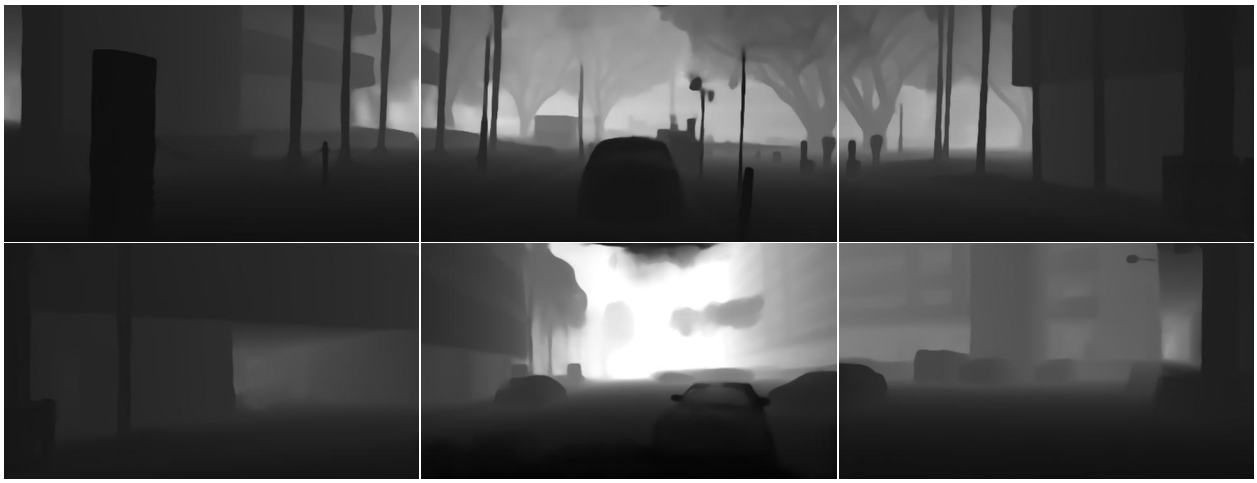

Figure 2. Visualization of scaled VGGT depth prediction on example frames.

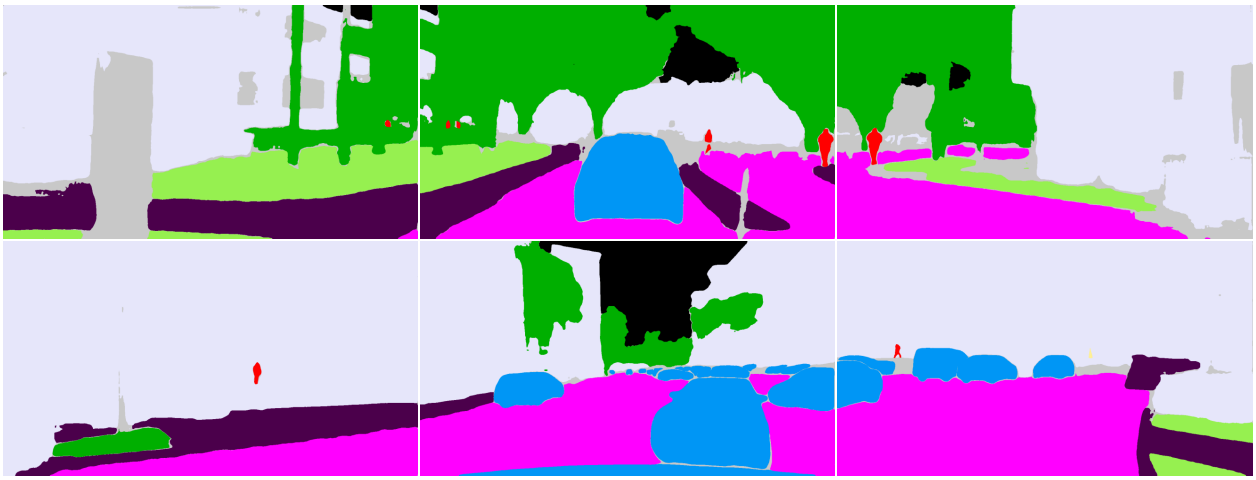

Figure 3. Visualization of OpenSeeD segmentation results on example frames.

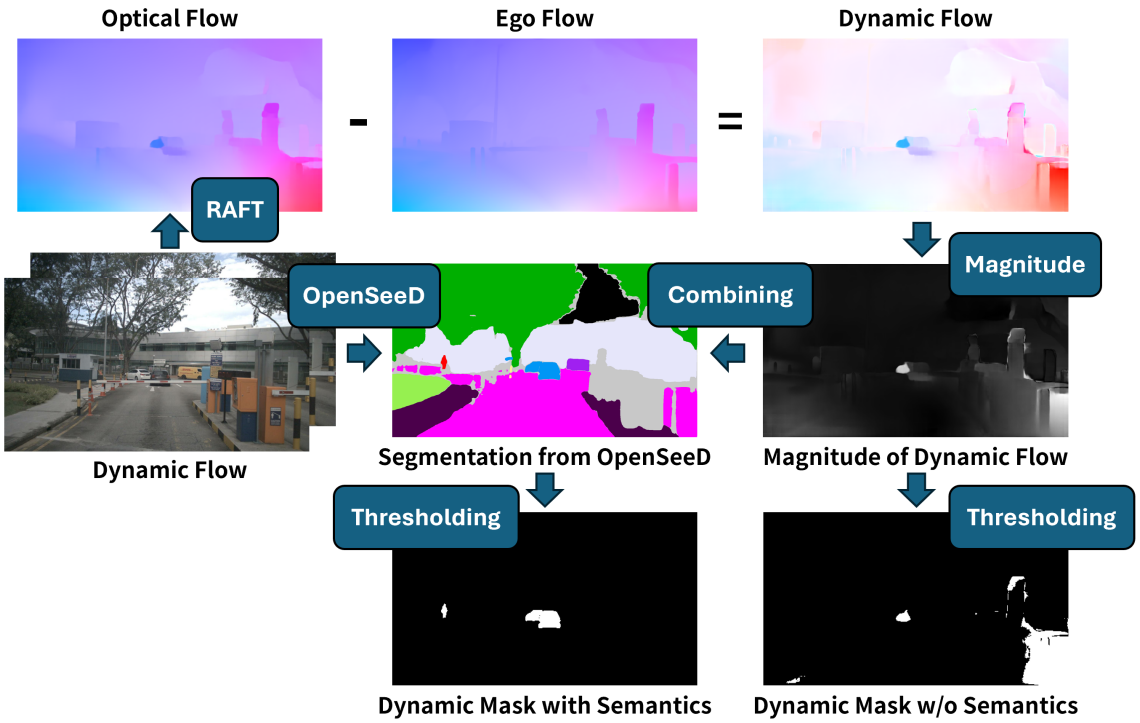

Figure 4. Illustration of the tracking process in TT-OccCamera.

each point with a specific foreground object. Due to the often imprecise boundaries of predicted masks, the resulting instance-level point sets can contain substantial noise. To address this, we apply DBSCAN clustering [3] to each instance's point cloud to extract its core structure and eliminate outliers. This approach proves effective in significantly removing noise, as illustrated in the left column of Fig. 5, where gray points are obtained by directing projecting onto segmentation masks and green points represent the denoised output after DBSCAN clustering (slightly translated for observation). We then perform object-level matching across adjacent frames based on the spatial proximity and shape similarity of the filtered point clusters. For each matched pair, the 3D flow is estimated using the Iterative Closest Point (ICP) algorithm [1]. Qualitative results are presented in the right column of Fig. 5, where green, blue, and red points represent the source points, destination points, and the ICP-transformed source points, respectively. Green arrows indicate the estimated 3D flow vectors. The effectiveness of the ICP-based alignment can be clearly observed. Finally, matched points are propagated to the next frame, while unmatched instances from the previous frame are discarded to avoid the accumulation of errors caused by moving or disappearing objects.

Table 1. **Robustness evaluation on rainy and nighttime nuScenes [2] scenes (mIoU).** R: rainy. N: nighttime. TT-Occ variants consistently outperform the task-specific SelfOcc [4] model across all challenging conditions.

Method	Scene ID 0911 (R)	0915 (R)	1065 (R+N)	1067 (R+N)	1073 (N)	Avg
SelfOcc	18.3	11.0	7.9	7.3	6.2	10.1
TT-OccCamera	21.2	13.6	11.2	9.4	10.3	13.1
TT-OccLiDAR	**27.0**	**18.9**	**12.0**	**13.9**	**16.2**	**17.6**

B. Additional Results

B.1. Performance under Challenging Conditions

A key strength of our system is that it is built upon large-scale foundation models, which are trained on diverse datasets and therefore exhibit strong generalization beyond specific domains. Leveraging these models gives TT-Occ a natural advantage over task-specific approaches such as SelfOcc [4], particularly under challenging conditions. To assess robustness, we evaluated both the camera- and LiDAR-based variants on nighttime and rainy scenes from the nuScenes [2] dataset. As shown in Table 1, TT-Occ consistently surpasses SelfOcc across all challenging scenarios, demonstrating improved accuracy and robustness.

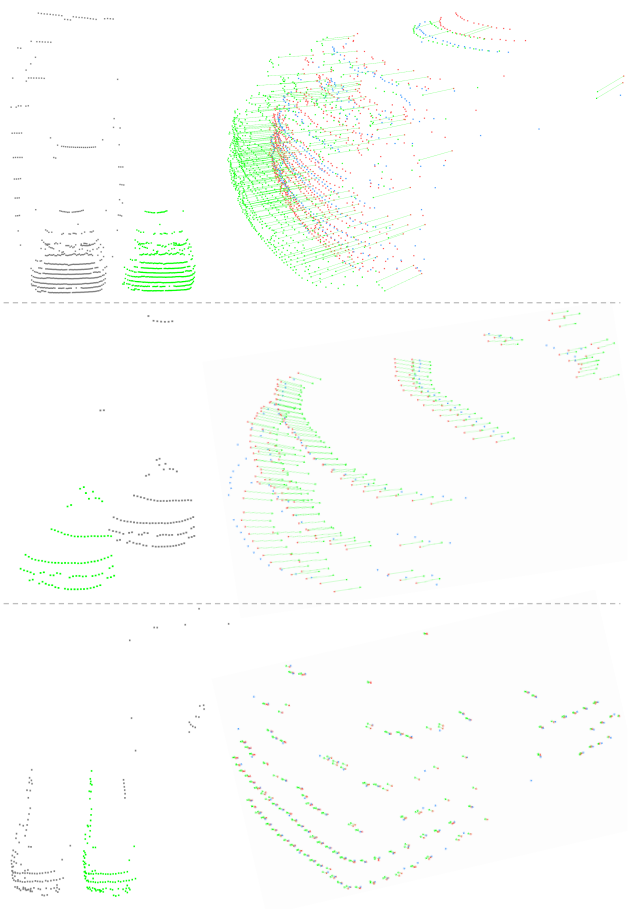

Figure 5. **Visualization of instance-level point cloud denoising and 3D flow estimation.** From the first to the third row, we show three example **car** instances from LiDAR data. Left: gray points represent the raw instance points obtained from segmentation masks, while green points are the core structures extracted via DBSCAN (visualized with a slight offset for clarity). Right: ICP-estimated 3D flow between adjacent frames, where green, blue, and red denote the source, target, and aligned source points, respectively. Green lines indicate the estimated flow vectors. It can be seen that DBSCAN effectively removes noisy outliers, and ICP produces accurate frame-to-frame alignment.

B.2. Expanded Ablation Comparisons

We further present enlarged visual comparisons of the variants evaluated in our ablation study. These visualizations reinforce the conclusions drawn in the main paper. Variant $\mathbb{A}$, where Gaussians are initialized using the "lift" strategy at each time step without temporal information, performs poorly due to sparse observations and the lack of anisotropic occupancy modeling needed to approximate local geometry. Introducing covariance-aware voxelization and scale regularization in $\mathbb{B}$ leads to consistent improvements across both static and dynamic classes for both LiDAR and camera settings. Allowing Gaussians to accumulate over time in $\mathbb{C}$ further boosts performance on static classes by aggregating evidence across frames, but severely degrades dynamic class accuracy due to untracked motion, resulting in trailing artifacts. Incorporating dynamic Gaussian tracking in $\mathbb{D}$ restores temporal consistency and substantially improves dynamic class performance while preserving strong performance on static content, producing clean and artifact-free occupancy.

In addition to these variants, we include baseline $\mathbb{E}$, which integrates the optional TRBF fusion module. Although $\mathbb{D}$ already handles dynamic objects effectively, we still observe scattered high-frequency noise, particularly in the camera variant, primarily due to segmentation boundary inaccuracies and imperfect dynamic region estimation. While this noise is extremely sparse and has negligible impact on overall accuracy, it slightly reduces visual quality. To mitigate this, TRBF fusion is applied as an optional spatio-temporal smoothing module. As shown in Fig. 6, TRBF in TT-OccCamera $\mathbb{E}$ effectively suppresses residual noise and produces smoother and more visually coherent reconstruction results.

References

[1] P.J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE TPAMI*, 14(2):239–256, 1992. 3

[2] Holger Caesar, Varun Bankiti, Alex H. Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *CVPR*, pages 11618–11628, 2020. 3

[3] Martin Ester, Hans-Peter Kriegel, Jörg Sander, and Xiaowei Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, page 226–231, 1996. 3

[4] Yuanhui Huang, Wenzhao Zheng, Borui Zhang, Jie Zhou, and Jiwen Lu. Selfocc: Self-supervised vision-based 3d occupancy prediction. In *CVPR*, pages 19946–19956, 2024. 1, 3

[5] Qing Jiang, Junan Huo, Xingyu Chen, Yuda Xiong, Zhaoyang Zeng, Yihao Chen, Tianhe Ren, Junzhi Yu, and Lei Zhang. Detect anything via next point prediction. *arXiv preprint arXiv:2510.12798*, 2025. 1

[6] Nikhil Keetha, Norman Müller, Johannes Schönberger, Lorenzo Porzi, Yuchen Zhang, Tobias Fischer, Arno Knapitsch, Duncan Zauss, Ethan Weber, Nelson Antunes, et al. Mapanything: Universal feed-forward metric 3d reconstruction. *arXiv preprint arXiv:2509.13414*, 2025. 1

[7] Zachary Teed and Jia Deng. Raft: Recurrent all-pairs field transforms for optical flow. In *ECCV*, pages 402–419, 2020. 1

[8] Jianyuan Wang, Minghao Chen, Nikita Karaev, Andrea Vedaldi, Christian Rupprecht, and David Novotny. Vggt: Visual geometry grounded transformer. In *CVPR*, 2025. 1

[9] Hao Zhang, Feng Li, Xueyan Zou, Shilong Liu, Chunyuan Li, Jianwei Yang, and Lei Zhang. A simple framework for open-vocabulary segmentation and detection. In *ICCV*, pages 1020–1031, 2023. 1

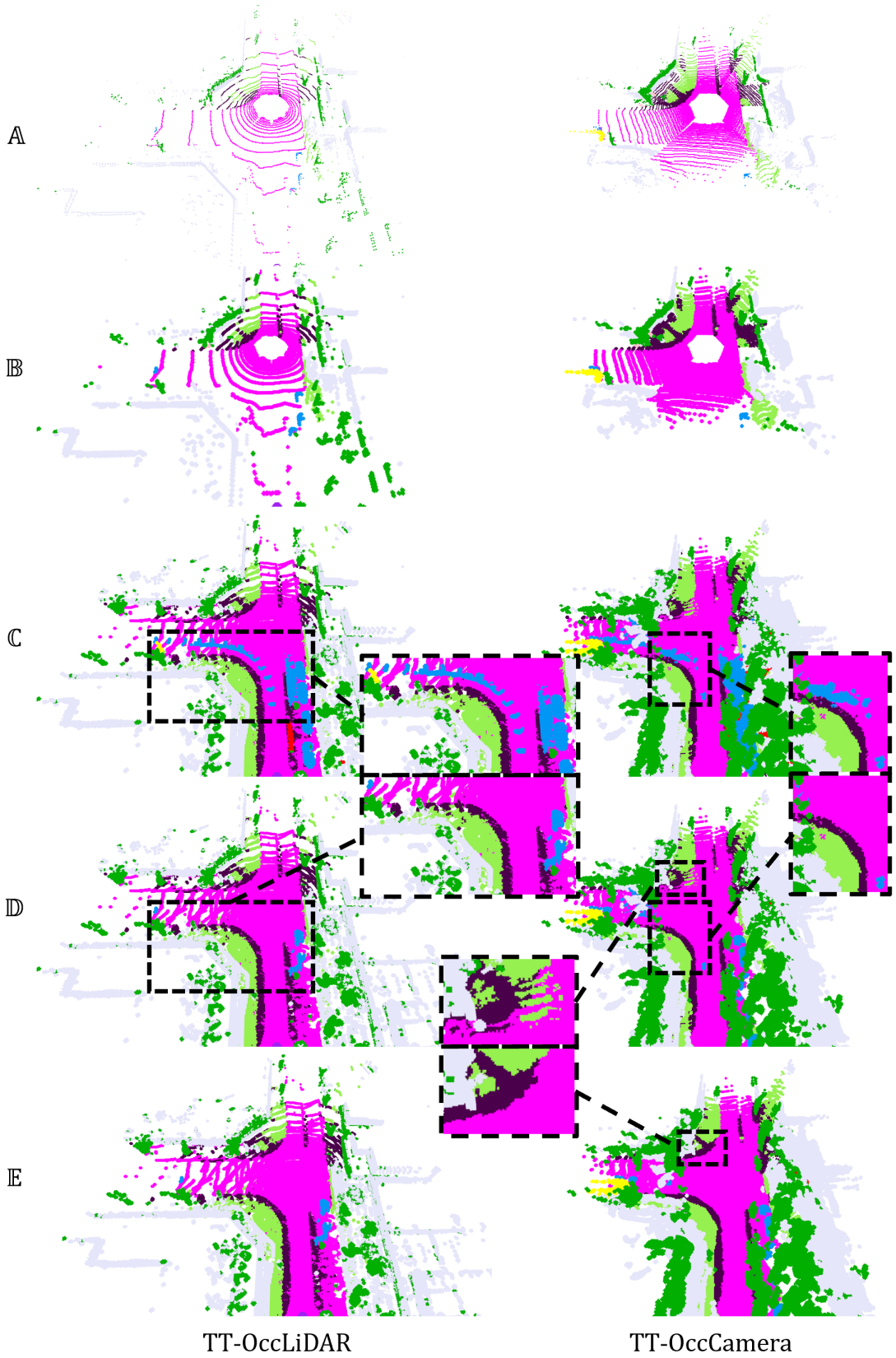

Figure 6. Zoomed-in visualization of different baselines of both variants of TT-Occ. 𝔸: Baseline. 𝔹: Covariance-aware Voxelization. ℂ: Inherit Previous Gaussians. 𝔻: Track Dynamic Gaussians. 𝔼: TRBF Fusion.